A Framework of Tool Integration for Internet-Based E-Commerce

Jianming Yong[1,2], Yun Yang[2]

[1]Department of Information Systems, Faculty of Business
University of Southern Queensland, Toowoomba, QLD 4350, Australia
yongj@usq.edu.au
[2] CICEC - Centre for Internet Computing and E-Commerce
School of Information Technology, Swinburne University of Technology
PO Box 218, Hawthorn 3122, Australia
yyang@it.swin.edu.au

Abstract. Tool integration has been an important research area for many years after software engineering became a key player in information industry. With the Internet establishing ubiquitous connectivity, more and more tools become reachable through the Internet. The Internet has massively changed the environment of software engineering. This paper extensively discusses relevant technologies of tool integration for Internet-based e-commerce. A new framework of tool integration for Internet-based e-commerce is proposed designed and demonstrated. An XML-based message server can effectively serve tool integration for Internet-based e-commerce within a business/integration domain.

1 Introduction

Tool integration has been researched for many years with the development of software engineering. Traditionally tool integration was implemented by an integrated environment, like Pact environment [1], Gravity [2], Shamash [3], PCE, the Field environment [4], the Forest environment , EAD environment, Integrated project-support environment, etc. But after the Internet has been widely accepted as a main vast platform for e-commerce, the traditional tool integration, which mainly focuses on the environment, has to be adjusted to meet the needs of Internet-based e-commerce. Thus tool integration will need to facilitate the Internet-based e-commerce. In recent years more research on tool integration is placed on the Web-related environments and applications. Actually the Internet has pushed software engineering into a more open-standard and distributed environment. That brings a big challenge to software tools, which contribute to the main development of software engineering. Especially in the recent years the World Wide Web has become an effective platform for e-commerce. Currently e-commerce falls into heavy Web-based applications, which will need the following requirements [5]:

- The necessity of handling both structured data and non-structured data
- The support of exploratory access through navigational interfaces
- A high level of graphical quality
- The customisation and possibly dynamic adaptation of content structure, navigational primitives, and presentation styles

- The support of proactive behaviour
- Security, scalability, and availability
- Interoperability with legacy systems and data
- Ease of evolution and maintenance

Based on these requirements, different kinds of tools will be needed to facilitate Internet-based e-commerce. Different tools, which may serve for a same or similar function, may be created by different vendors. Even same vendors may have different tools to serve different purposes over the Web-based applications, especially e-commerce. It is very important to find a way to effectively integrate tools, which are used to create applications or tools over the Internet. Thus tool integration over Internet-based e-commerce becomes a very challenging task for researchers.

Although tool integration cannot completely overcome all the problems and the inconsistence among software tools will exist forever, this paper will try to discuss some important issues related to tool integration over Internet-based e-commerce. Through this, a mechanism can be established to better serve the development of Internet-based e-commerce from the perspective of tool integration.

This paper will be organised as follows. Section 2 will discuss the application architecture of Internet-based e-commerce. Section 3 will introduce some tools for Internet-based e-commerce. Section 4 will review some methods of tool integration. Section 5 will propose a tool integration framework for Internet-based e-commerce. Sections 6 and 7 will detail the design and demonstration of tool integration for Internet-based e-commerce. Section 8 will conclude this paper.

2 Application Architecture of Internet-Based E-Commerce

In order to better understand how the Internet facilitates the e-commerce transactions, we use a traditional way, like the network application architecture, to split functions, which usually are created by tools, between clients (customers) and servers (sellers/service providers). Thus the following architecture can be used to describe how the functions of e-commerce can be achieved by client-server e-commerce systems.

2.1 Client-Server E-Commerce Systems

Today most organisations are using client-server architectures to build their applications and services. This architecture overcomes the shortcomings of the pure client-based or server-based architectures in isolation and can flexibly balance the processing between the client and the server based on the property of applications. Thus this architecture can satisfy the requirements of Internet-based e-commerce. The functions and tools are distributed among clients and servers based on business requirements. This architecture can be flexibly tailored to thin-clients or fat-clients according to different requirements of different e-commerce systems. For example, if clients are only the customers of this system, in the client side, minimal functions and tools will be required to sit on the client side. But if the clients are the developers of e-commerce systems, which may require more functions and tools to sit on the client side so that the clients have enough capacity to required e-commerce functions, which will be provided to the customers over the Internet. Although this architecture can better serve for Internet-based e-commerce systems than other two architectures, it is not easy for tool integration because different vendors may have quite different client systems and server systems, which support by broad tools. In order to facilitate tool integration, traditional two-tier client-server architecture needs to extend to multi-tier client-server architecture.

3 Tools for Internet-Based E-Commerce

Because Internet-based e-commerce has massively involved the Web usage, tools related to Web development have played a big role in Internet-based e-commerce. As Fraternali [5] pointed out that existing tools have been grouped into six categories for Web tools:

- Visual editors and site managers;
- Web-enabled hypermedia authoring tools;
- Web-DBPL integrators
- Web form editors, report writers, and database publishing wizards;
- Multi-paradigm tools;
- Model-driven applications generators

In addition to previous six categories of Web application tools, Internet-based e-commerce definitely needs all sorts of security tools to support its business implementation and transactions. Well-known security tools include networking monitoring tools, authentication/password tools, service-filtering tools, tools to scan hosts for known vulnerabilities, integrity-checking tools, encryption tools, etc. Only with these security tools help, Internet-based e-commerce can be broadly conducted by the convinced producers and customers.

4 Traditional approaches for tool integration

There are several traditional approaches for tool integration: tool-to-framework integration [6], tool-to-tool integration [7], tool integration based on the message server [8].

Tool-to-framework integration adopts an approach that tools can be encapsulated within a single software engineering framework, with all communication taking place by way of the framework. This approach is very feasible within one organisation/Intranet. But because of the diversity of the Internet, which connects countless heterogeneous systems, which are related to different software engineering frameworks, this approach becomes impossible for Internet-based e-commerce.

Tool-to-tool integration is based on the belief that tools may also be integrated directly, with some communication by-passing the underlying framework. It works very well for certain scenarios. For example, if one tool has a fixed connection with another tool, we can directly use some techniques to integrate two tools. But for Internet-based e-commerce systems, most connections are not fixed and also so many tools are dispersed over the Internet. It is almost impossible to integrate those countless tools together based on the one-by-one method. Thus this approach cannot satisfy the requirements of tool integration for Internet-based e-commerce.

Tool integration based on the message server was initially proposed by Reiss [4]. Its prototype is called Field. The distinguishing feature of Field is a centralised message server, called Msg, which routes messages between tools. Each tool in the environment registers with Msg a set of message patterns that indicate the kinds of messages it should receive. Tools send messages to Msg to announce changes that other tools might be interested in. Msg then the server “selectively” broadcasts those messages to tools whose patterns match those messages. Figure1 shows how Field works.

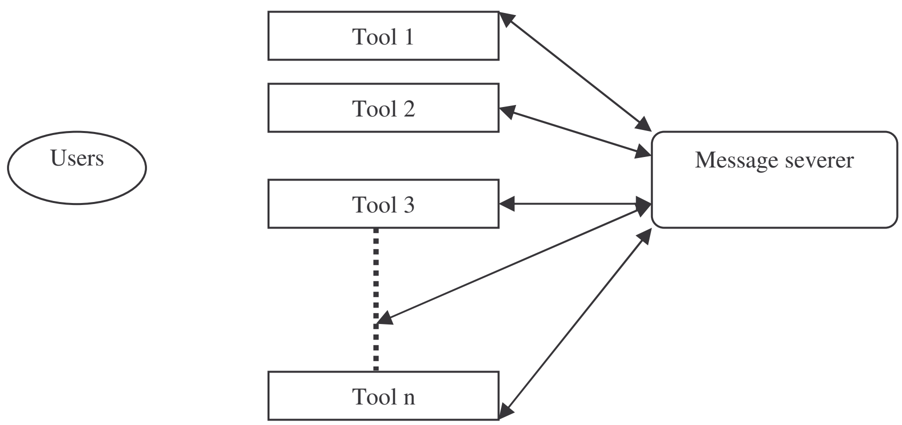

Figure 1. Message server working with tools

For example, a user makes a change in Tool 1. Tool 1 will tell the message server what has changed. Then the message server broadcasts the change to all associated tools. Suppose Tool 3 is interested in this change because this change will affect tool 3 later on. Thus tool 3 adjusts its functions to respond to Tool 1’s change. But this broadcast-based tool integration does not solve the problem of the control issues. For example, after Tool 1 notifies the change, the message server will broadcast this change to all other members. Then no one knows what will happen to related tools. Thus there will need a control mechanism to know what should happen. In order to solve this issue, the Field system needs extending the policy support for selective broadcast in tool integration (Figure 2).
The policy repository has the rules that determine how and when tools are invoked in the software development process. It still uses a centralised message server. But the server maintains a set of patterns that indicate messages of interest for each tool. The action will be taken only when a message is successfully matched against a pattern by a set of condition-action pairs. These pairs define the interaction policy for that tool by controlling which messages are routed between tools.

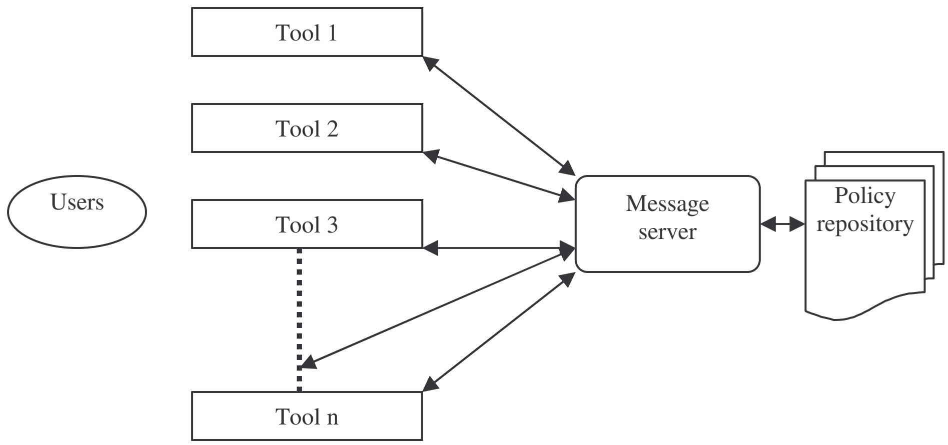

Figure 2. Policy support for selective broadcast

5 New Integration Framework for Internet-Based E-Commerce

Because the Internet has the characteristics of broad distribution, and also all heterogeneous business domains, entities, processes are connected together by the Internet, it is impossible to use one standard/approach/theory to deal with tool integration for Internet-based e-commerce. Traditional tool integration mechanisms, which mainly focus on creating a relatively unified environment, such as PSE (programming support environments) and SEE (software engineering environments) [9], cannot satisfy this demand, which has taken place among business environments because of the Internet use. In order to adopt and adapt to this ubiquitous connection by the Internet, a new methodology for tool integration has been proposed to deal with tool integration for Internet-based e-commerce. Based on the current popular architecture, which is discussed in the architecture section (section 2), of Internet-based e-commerce, we propose a ubiquitous tool integration framework for Internet-based e-commerce, distributed message connection through middleware server with XML. Because XML actually has been accepted as a de facto standard for message descriptions and exchanges, which are understandable to both human and computers, a significant convenience is brought into message-based tool integration for Internet-based e-commerce. A new framework is illustrated in Figure 3.

If one tool wants to communicate with any other tools, it has to send an XML message to the message server. The message server needs to read this message against its message repository (MR). Then after the message matches its MR, the action rule repository (ARR) will be searched to use relevant rules to guide which actions should be taken by relevant tools. The details are in the next section.

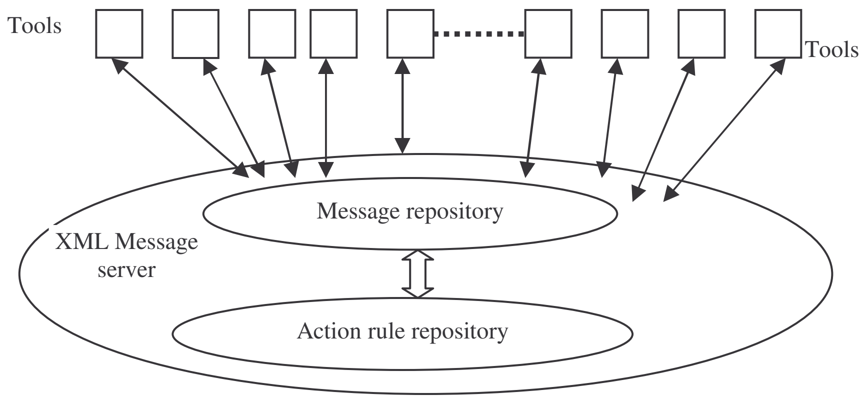

Figure 3. Framework of XML-message-based tool integration

6 Design of XML-Message-Based Tool Integration

Because the XML MR needs to feeding in XML messages, all relevant tools are required to have ability to output XML messages for the message server to accept. Most current tools have XML message output functions. But some “legacy” tools cannot provide the XML message output. They need translation agents to facilitate this connection. Because the research on agents is beyond the scope of this paper, it will not be detailed how translation agents work for “legacy” tools and the message server. Instead, MR and ARR are the key to tool integration for Internet-based e-commerce. Thus details of MR and ARR are discussed in this sections.

6.1 Message Repository (MR)

MR has a full list of all tools which belong to one integration domain. In most cases, one tool usually only needs cooperation from other tools within one integration domain. The MR is organised as a tree structure, which is easy for an incoming message to match its desired pattern.

The document head includes information about the XML version, encoding, name space of integration domain, etc. All XML message blocks have XML schemas for activating incoming messages from requested tools. If a message is for Tool 1, the message will be sent to Tool 1 XML message block for further processing, which is relevant to ARR. If one tool needs cooperation beyond its integration domain, the message server will need to negotiate with other message servers to facilitate this cooperation. Then the message will be sent to the outside domain XML message block for further processing.

6.2 Action Rule Repository (ARR)

After incoming messages are matched in MR, MR will request ARR to retrieve related action rules for requested tools to take actions in order to achieve the purpose of integration. ARR has all designed rules.

7. Demonstration

In order to clearly demonstrate how MR and ARR work together, a finite state machine (FSM) is used to model their transactions. Suppose there are a number of tools in the integration domain. These tools are modeled as a set: T(t1, t2, …, tn). T sends out messages for MR' FSM to process. After MR's FSM accepts incoming XML messages, ARR's FSM is activated. This process is shown in Figure 4.

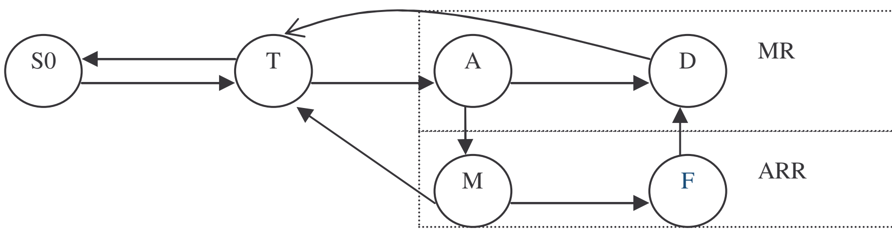

Figure 4. FSM in message server

- (S0, users)→T
  S0 is an initial state of the system, no tools need cooperation from others. After users start a tool for a request of cooperation from others, state S0 transits into state T.
- (T, XML messages)→A
  When FSM stays at state T and any XML messages send out, state T will transit into state A, which is located at MR.
- (A, matched messages)→M
  When FSM stays at state A and matched XML messages are found, state A will transit into state M, which is located at ARR.
- (A, non-matched messages)→D
  When FSM stays at state A and no XML messages are matched in MR, state A will transit into state D, which is located at MR for abnormal processing.
- (M, satisfied conditions)→T
  When FSM stays at state M and satisfied conditions are identified, state M will transit into state T, which notifies other tools to take action to cooperate with the initial tool. If those woke-up tools need further cooperation, a new cycle of T→A→M→T will happen.
- (M, unsatisfied conditions)→F
  When FSM stays at state M and no satisfied conditions are found, state M will transit into state F, which is in charge of abnormal processing.
- (F, error process) →D
  When FSM stays at state F and error processes have been finished, state F will transit into state D.
- (D, error report) →T
  When FSM stays at state D and error reports return to initial tools, state D will transit into state T for another new cycle.

- (T, no new inputs) →S0
  When FSM stays at state T and no any new events happen, state T will transit into state S0, which the system is in the idle status.

8 Conclusions

This paper extensively reviews tool integration from the perspective of software engineering. Then tool integration for Internet-based e-commerce is broadly discussed. Many useful Web tools are categorised in the paper. Finally a new framework of tool integration for Internet-based e-commerce has been proposed. Furthermore the design has been illustrated with a finite state machine used to demonstrate the transaction between tools and the message server as proof of concept, which is the key of this integration solution. Through the new framework of tool integration for Internet-based e-commerce, more and more Internet-based tools can be effectively integrated with each other and serve better for business functions.

References

1.Thomas, I. Tool Integration in the Pact Environment. Proc. of the 11th international conference on Software engineering. 1989. Pittsburgh, Pennsylvania, United States: ACM, pages:13-22.

2.Rangarajan, M., et al. Gravity: An Object-Oriented Framework for Hardware/Software Tool Integration. Proc. of the 30th annual Simulation Symposium (SS'97). 1997. Atlanta, GA: IEEE, pages:24-30.

3.Camacho, D., et al. Shamash: An AI tool for Modelling and Optimizing Business Processes. Proc. 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01). 2001. Dallas, Texas, USA: IEEE, pages:306-313.

4.Reiss, S.P., Connecting Tools Using Message Passing in the Field Program Development Environment. IEEE Software, 1990. **7**(4): p. 57-66.

5.Fraternali, P., Tools and Approaches for Developing Data-intensive Web Applications: a Survey. ACM Computing Surveys (CSUR), 1999. **31**(3): p. 227-263.

6.Gautier, B., et al. Tool Integration: Experiences and Directions. Proc. of the 17th international conference on Software engineering. 1995. Seattle, Washington, United States: ACM, pages:315-324.

7.Bredenfeld, A. and R. Camposano. Tool Integration and Construction Using Generated Graph-based Design Representations. Proc. of the 32nd ACM/IEEE conference on Design automation conference. 1995. San Francisco, California, United States: ACM Press New York, NY, USA, pages:94-99.

8.Garlan, D. and E. Ilias. Low-cost, Adaptable Tool Integration Policies for Integrated Environments. Proc. of the fourth ACM SIGSOFT symposium on Software development environments. 1990. Irvine, California, United States: ACM Press New York, NY, USA, pages:1-10.

9.Ossher, H., W. Harrison, and P. Tarr. Software Engineering Tools and Environments: a Roadmap. Proc. of the conference on The future of Software engineering. 2000. Limerick, Ireland: ACM Press New York, NY, USA, pages:261-277.